

Nuclear Fuel Cycle Workshop Frameworks Discussion (Group 2A)

Grand Challenges for Frameworks

- **Earn acceptance** for large-scale simulation tools in the nuclear industry
- **Provide tools and techniques** to enable nuclear industry applications to exploit large-scale simulation
- **Work with end users** to improve the state-of-the-practice of integrated modeling
- **Raise human productivity and execution efficiency** for scientific discovery and engineering design

Grand Challenge 1

- **Earn acceptance** for large-scale simulation tools in the nuclear industry
 - Large potential advantage needed to justify conversion
 - ASC and SciDAC programs show that large-scale simulation can revolutionize engineering and science R&D programs
 - In expected ways: focusing investments in expensive and high latency experimental programs with cheaper and quicker exploration of parameter space through simulation
 - In unexpected ways: forcing re-examination of trusted empiricism and driving simulation breakthroughs through new challenges

Grand Challenge 2

- **Provide tools and techniques** to enable nuclear industry applications to exploit large-scale simulation
 - Through well-defined, thoughtfully negotiated interfaces that separate concerns and provide stability as
 - architectures evolve
 - application needs evolve
 - computational capabilities improve
 - With portability across architectural extremes
 - From laptop to petaflop

Grand Challenge 3

- **Work with end users** to improve the state-of-the-practice of integrated modeling
 - Encourage relaxation of assumptions and lumped-parameter models and greater reliance on more first-principles models, to harness growing capabilities of computation
 - Lower the expertise threshold required to employ “best practices and software”

Grand Challenge 4

- **Raise human productivity and execution efficiency** for scientific discovery and engineering design
 - Reduce lead times for deployment of next-generation plants
 - Reduce risk in expensive software development
 - Help create “born verified” code

Major Issues

- Standard interface definitions – a joint responsibility of analysts and math/CS developers
 - Identify the abstractions required and provide them at multiple levels
- Interaction of frameworks used in nuclear fission power R&D with export control restrictions
 - Finer control through modularity
- Interaction of code frameworks use in nuclear fission power R&D with licensing/certification (not necessary for *all* codes)
 - Streamlined approvals through reuse of accepted components

Relevant application domains

- Materials/fuels properties, phase diagrams
- Reactor modeling
 - Fluid/thermal
 - Radiation transport
 - Structural
- Reactor control and safety
- Reprocessing/separation
- Repository modeling
- Infrastructure modeling (systems analysis, including security, non-proliferation, and economics)

Enabling Math Technologies

- Continuous models (PDEs, IEs)
 - discretizations, stiff integrators, solvers, etc.
- Discrete models (MD, Kinetic, etc.)
- Optimization
 - parameter identification, control, design, etc.
- Uncertainty modeling, sensitivity analysis
- Model management
 - CAD to mesh, adaptivity, dynamic load balancing, etc.
- Code coupling techniques
 - implicit algorithms and software tools encouraging legacy code reuse
- Multiscale methods, homogenization techniques
- Reduced-order modeling

Enabling CS Technologies

- Standards for parallel programming (MPI, etc.)
- Standard interfaces for kernel algorithms (linear solvers, etc.)
- Language interoperability, componentization
- Visualization
- High-performance I/O
- Performance monitoring/debugging
- Distributed, associatively labeled data archiving
- Automated discovery tools/data mining
- Assimilation and fusion of experimental data with simulations

Taxonomy of Framework Use

- “Data backplane” framework, e.g., “plasma state” in the FSP program
- “Toolkit” (library-like) use of infrastructure (meshers, solvers, partitioners, visualizers, data management systems, I/O, etc.) within a single scaled application
- “Dakota-like” framework for outer loop analyses
- “Loosely coupled” frameworks for multiphysics simulations
- “Tightly coupled” frameworks for multiphysics simulations

Key “salesmanship” issues

- “Before and after” scenarios
 - 2D lumped parameter reactor models versus 3D (more) first-principles models
 - Heroic one-off repository simulations versus exploring large design space of scenarios incorporating uncertainties
- Rapid response scenario
- Economic impact evaluation

Challenges for math software designers

- The past challenge:
Increase functionality and capability for a small number of users who are expert in the domain of the software
- A future challenge:
Increase ease of use (for correctness *and* for efficiency) for a large number of users who are expert in something else

Design principle: multiple layers

- Top layer (all users)
 - Abstract interface featuring language of application domain, hiding details, with conservative parameter defaults
 - *Robustness, correctness, ease of use*
- Middle layers (experienced users)
 - Rich collection of state-of-the-art methods and data structures, exposed upon demand, highly configurable
 - *Capability, algorithmic efficiency, extensibility, composability, comprehensibility of performance and resource use*
- Bottom layer (developers)
 - Support for variety of execution environments
 - *Portability, implementation efficiency*

What engineers want in math software

- Develop code “without having to make bets”
 - accomplish certain abstract mathematical tasks
 - stay agnostic about particular solution methods and codes
 - run on laptops (on travel), low-cost unmetered clusters (at work), and on unique shared national resources
- Ordered goals (need them all for production use)
 - usability and robustness
 - portability
 - algorithmic efficiency (optimality) and implementation efficiency (within a processor and in parallel)
- Algorithmic optimality and software stability
 - scalable methods needed for multiscale problems
 - no “hand coding” for evanescent environments

Foci for math ISICs

- The past focus:
Meet the understood and expressed needs of application groups, where they are, typically across a conventional interface
- A future focus:
Lure application groups into new explorations, boldly going where no one has gone before, typically blurring conventional interfaces

Math ISICs

APDEC

Applied Partial Differential Equations Center, *directed by
P. Colella*



Terascale Simulation Tools & Technologies,
directed by D. Brown, L. Diachin, J. Glimm



Terascale Optimal PDE Simulations
directed by D. Keyes